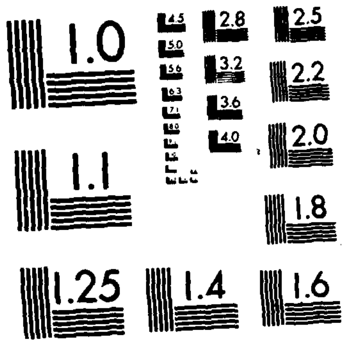
AD-A134 819 ANALYSIS OF AN ERROR DETECTION SCHEME FOR PARALLEL COMPUTATIONS(U) JOHNS HOPKINS UNIV BALTIMORE MD DEPT OF ELECTRICAL ENGINEERIN.. G G MEYER ET AL. 15 NOV 83 1/1

UNCLASSIFIED JHU/EECS-83/14 N00014-81-K-0813 F/G 9/4 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

The Johns Hopkins University

AD-A134819

ELECTRICAL ENGINEERING & COMPUTER SCIENCE

ANALYSIS OF AN ERROR DETECTION SCHEME FOR PARALLEL COMPUTATIONS

Gerard G. L. Meyer and Howard L. Weinert

Report JHU/EECS-83/14

Electrical Engineering and Computer Science Department

The Johns Hopkins University

Baltimore, Maryland 21218

This work was supported by the Office of Naval Research under Contract N00014-81-K-0813.

ABSTRACT

~~We~~ determine the conditions under which an error detection scheme based on strict redundancy can be used to increase confidence in the results of parallel computations. This study shows that the issues of speed and reliability of parallel processors are interdependent and must be considered jointly at the design stage.

1. INTRODUCTION

Recent technological developments have made parallel processing a viable option for achieving desired computational speed. Given a problem of interest and an associated procedure to obtain its solution, a cluster of β computing elements may be used to produce the solution in the required time, provided that an appropriate decomposition of the procedure can be found. For real-time processing problems where the speed constraints may be quite severe, the number of computing elements required can be large. Besides the difficulty associated with the decomposition of the solution procedure, the use of a large number of computing elements introduces a new set of problems. In particular, the probability that all computing elements produce correct results becomes vanishingly small as the number of computing elements increases. Thus, the computing cluster may produce a result in the required time, but the probability that the result is also the solution of the problem of interest decreases with the number of computing elements in the cluster. It is clear, therefore, that the issues of speed and reliability are interdependent and cannot be treated separately.

The quantity that characterizes the reliability of a computing cluster is P_C, the probability that the output of the cluster is correct. In order to analyze this and other quantities introduced later on, we shall adopt the following hypothesis:

<u>Hypothesis 1</u>:

(i) the input to the cluster is correct,

(ii) each computing element in the cluster has the same probability p of being non-faulty,

(iii) the computing elements fail independently.

Therefore, if all the computing elements in a cluster are fault-free, the cluster output will be correct. The converse is not necessarily true, since a computing element may be faulty without affecting the cluster output. It follows from Hypothesis 1 that a lower bound for P_C is given by

$$P_C \geq p^{\beta} = P_{C,m}. \tag{1}$$

The quantity p depends on the type of computing element used and on the time interval over which we are interested in the output. This means that all the probabilistic quantities discussed in this paper are with reference to the same time interval. For example, if p is the probability that a computing element remains non-faulty for twenty-four hours, then $P_{C,m}$ is a lower bound on the probability that the output of a computing cluster is correct over the same twenty-four hour period.

Given p and β, the value of $P_{C,m}$ may not be large enough for our purposes, and therefore our confidence in obtaining the correct result will not be high enough. One way to increase our confidence in the correctness of the output is to try to detect output errors, and then to accept the output only when no error has been detected. In this case, there are two quantities of interest: the probability that the output of the original cluster is correct given that we accept it, and the proba-

bility that we reject the output of the original cluster given that it is correct (false alarm).

While an off-line fault detection scheme might allow us to ascertain that no hardware fault is present during the application of the tests, we would have no assurance that the cluster remains non-faulty during the actual computation of the desired result. Since we are interested in the correctness of the results produced by the computing cluster and not in the possible existence of hardware faults, and since we cannot ascertain the correctness of the results before they are produced, we therefore need a concurrent error detection scheme. One way to implement such a scheme is to use strict redundancy: replicate the initial computing cluster (CC_1) $\alpha-1$ times, send the original input to all the clusters (CC_1, CC_2,..., CC_α), and then compare their outputs in order to produce a boolean variable b that equals zero if all cluster outputs are identical, and that equals one otherwise. If $b = 0$, we accept the output of CC_1; if $b = 1$, we reject it. This approach is appealing because, once a computing cluster that meets the computational speed requirements has been designed, the replication does not involve any additional design effort.

Although similar error detection schemes have been used in various contexts for some time (see [AVI78] and the references therein), no complete analysis of their usefulness under any reasonable set of assumptions has been carried out. Such an analysis is presented in Section 2, and the consequent implications for reliable parallel processing are detailed in Section 3.

2. ANALYSIS OF THE ERROR DETECTION SCHEME

In this paper, we shall assume that the error detector is always non-faulty. In other words, we shall adopt the following hypothesis:

Hypothesis 2: The detector produces $b = 0$ if and only if all cluster outputs are identical.

Let $P_{CD}(\alpha,\beta)$ be the probability that the output of CC_1 is correct given $b = 0$, and let $P_{FA}(\alpha,\beta)$ be the probability that $b = 1$ given that the output of CC_1 is correct. Lower and upper bounds for $P_{CD}(\alpha,\beta)$ and $P_{FA}(\alpha,\beta)$, respectively, will now be derived.

Given α clusters, each containing β computing elements, let $E_j(\alpha,\beta)$ be the event that exactly j clusters are faulty. A cluster is faulty if and only if at least one of its elements is faulty, and therefore the probability that a cluster is faulty is $1-p^{\beta}$. As a result,

$$P(E_j(\alpha,\beta)) = \frac{\alpha!}{j!(\alpha-j)!}(1-p^{\beta})^j p^{\beta(\alpha-j)}, \quad j = 0,1,\ldots,\alpha. \tag{2}$$

Let $B_j(\alpha,\beta)$ be the event that exactly j cluster outputs are incorrect. It is clear that if no more than k clusters are faulty, then at most k cluster outputs can be incorrect, and if at least k cluster outputs are incorrect, then at least k clusters are faulty. Thus

$$\sum_{j=0}^{k} P(B_j(\alpha,\beta)) \geq \sum_{j=0}^{k} P(E_j(\alpha,\beta)), \tag{3}$$

and

$$\sum_{j=k}^{\alpha} P(B_j(\alpha,\beta)) \leq \sum_{j=k}^{\alpha} P(E_j(\alpha,\beta)). \tag{4}$$

Lemma 1: Under Hypotheses 1 and 2,

$$P_{CD}(\alpha,\beta) \geq \frac{p^{\beta\alpha}}{p^{\beta\alpha} + (1-p^{\beta})^{\alpha}} = P_{CD,m}(\alpha,\beta). \tag{5}$$

Proof: By definition

$$P_{CD}(\alpha,\beta) = P(CC_1 \text{ output correct} \mid b=0)$$

$$= \frac{P(CC_1 \text{ output correct and } b=0)}{P(b=0)} .$$

Using Hypothesis 2, it is clear that

$$P(CC_1 \text{ output correct and } b=0) = P(B_0(\alpha,\beta)),$$

$$P(b=0) = P(B_0(\alpha,\beta) \text{ and } b=0) + P(B_\alpha(\alpha,\beta) \text{ and } b=0),$$

$$P(B_0(\alpha,\beta) \text{ and } b=0) = P(B_0(\alpha,\beta)),$$

and thus

$$P_{CD}(\alpha,\beta) = \frac{P(B_0(\alpha,\beta))}{P(B_0(\alpha,\beta)) + P(B_\alpha(\alpha,\beta) \text{ and } b=0)}. \tag{6}$$

Now

$$P(B_\alpha(\alpha,\beta) \text{ and } b=0) \leq P(B_\alpha(\alpha,\beta)),$$

and (3) and (4) imply

$$P(B_0(\alpha,\beta)) \geq P(E_0(\alpha,\beta)),$$

$$P(B_\alpha(\alpha,\beta)) \leq P(E_\alpha(\alpha,\beta)).$$

Therefore

$$P_{CD}(\alpha,\beta) \geq \frac{P(E_0(\alpha,\beta))}{P(E_0(\alpha,\beta)) + P(E_\alpha(\alpha,\beta))} = \frac{p^{\beta\alpha}}{p^{\beta\alpha} + (1-p^\beta)^\alpha}. \quad \square$$

The behavior of the lower bound $P_{CD,m}(\alpha,\beta)$ as a function of α is characterized in the following lemma.

Lemma 2: Suppose that Hypotheses 1 and 2 are satisfied. If $p^\beta > 0.5$, then $P_{CD,m}(\alpha,\beta)$ converges to one strictly monotonically as α goes to infinity. If $p^\beta = 0.5$, then $P_{CD,m}(\alpha,\beta) = 0.5$ for all $\alpha \geq 1$. If $p^\beta <$ 0.5, then $P_{CD,m}(\alpha,\beta)$ converges to zero strictly monotonically as α goes to infinity.

Proof: Let $q = p^\beta$. Then (5) can be written as

$$P_{CD,m}(\alpha,\beta) = \frac{q^\alpha}{q^\alpha + (1-q)^\alpha} = \frac{1}{1 + ((1-q)/q)^\alpha}.$$

If $p^\beta > 0.5$, then $(1-q)/q < 1$ and $P_{CD,m}(\alpha,\beta)$ is clearly a strictly increasing function of α with a limiting value of one. If $p^\beta = 0.5$, then $(1-q)/q = 1$ and obviously $P_{CD,m}(\alpha,\beta) = 0.5$ for all $\alpha \geq 1$. If $p^\beta <$ 0.5, then $(1-q)/q > 1$ and $P_{CD,m}(\alpha,\beta)$ is a strictly decreasing function of α with a limiting value of zero. $\square$

Lemmas 1 and 2 together with Equation (1) imply that (i) if $p^\beta >$

0.5, then $P_{CD,m}(\alpha,\beta) > P_{C,m}$ for all $\alpha \geq 2$; (ii) if $p^{\beta} = 0.5$, then $P_{CD,m}(\alpha,\beta) = P_{C,m}$ for all $\alpha \geq 2$; and (iii) if $p^{\beta} < 0.5$, then $P_{CD,m}(\alpha,\beta) < P_{C,m}$ for all $\alpha \geq 2$. It follows then that the error detection scheme can be used to increase our confidence in the output if and only if $p^{\beta} >$ 0.5. We note that in work done in the Fifties on constructing reliable logic devices from unreliable components, von Neumann recognized that redundancy can degrade overall performance unless the components have some minimum reliability (see [NEU63, pp. 305-306, 322-324, 329-378]).

The false alarm probability will be considered next.

Lemma 3: Under Hypotheses 1 and 2,

$$P_{FA}(\alpha,\beta) \leq 1 - p^{\beta(\alpha-1)} = P_{FA,M}(\alpha,\beta). \tag{7}$$

Proof: By definition

$$P_{FA}(\alpha,\beta) = P(b=1 \mid CC_1 \text{ output correct})$$

$$= 1 - P(b=0 \mid CC_1 \text{ output correct})$$

$$= 1 - \frac{P(b=0 \text{ and } CC_1 \text{ output correct})}{P(CC_1 \text{ output correct})}.$$

Now

$$P(b=0 \text{ and } CC_1 \text{ output correct}) = P(\text{all cluster outputs correct})$$

$$= \prod_{j=1}^{\alpha} (CC_j \text{ output correct})$$

and therefore

$$P_{FA}(\alpha,\beta) = 1 - \prod_{j=2}^{\alpha} P(CC_j \text{ output correct}).$$

If CC_j is non-faulty, its output will be correct, and thus

$$P_{FA}(\alpha,\beta) \leq 1 - \prod_{j=2}^{\alpha} P(CC_j \text{ non-faulty}) = 1 - p^{\beta(\alpha-1)}. \ \square$$

Note that $P_{FA,M}(\alpha,\beta)$ goes to one as α - s to infinity. Thus, if $p^{\beta} > 0.5$, we can insure that $P_{CD}(\alpha,\beta)$ is as -se to one as we wish by choosing a sufficiently large value for α, concomitant drawback is that $P_{FA}(\alpha,\beta)$ may also be close to one. In other words, if we accepted the output it would almost certainly be correct, but we would almost never accept a correct output.

3. EFFICIENT COMPUTING NETWORK DESIGN

Suppose that that it is possible to solve a given problem in a given time using a cluster of β computing elements of a given type T, and assume that the basic reliability p of a type T computing element is known. Furthermore, suppose that we want to design a computing network consisting of one or more such clusters so that an accepted output is correct with probability at least θ_1 $(0 < \theta_1 < 1)$. To satisfy this requirement, it is sufficient, in view of Lemma 1, to ensure that

$$\frac{p^{\beta\alpha}}{p^{\beta\alpha} + (1-p^{\beta})^{\alpha}} \geq \theta_1 \tag{8}$$

for some integer $\alpha \geq 1$.

Given p, β and θ_1, it may or may not be possible to satisfy (8) with some integer $\alpha \geq 1$. If (8) can be satisfied, the most efficient design is obtained when α is chosen to be the smallest integer α_* that satisfies (8). In order to analyze the feasibility and efficiency issues, we partition the set of all pairs (θ_1, p^{β}) into disjoint subsets R_0, R_1 and R_2 as follows:

$$R_0 = \{(\theta_1, p^{\beta}) \mid p^{\beta} < \theta_1 \leq 0.5\} \cup \{(\theta_1, p^{\beta}) \mid p^{\beta} \leq 0.5 < \theta_1\},$$

$$R_1 = \{(\theta_1, p^{\beta}) \mid p^{\beta} \geq \theta_1\},$$

$$R_2 = \{(\theta_1, p^{\beta}) \mid 0.5 < p^{\beta} < \theta_1\}.$$

<u>Lemma 4</u>: If (θ_1, p^β) is in R_0, then (8) cannot be satisfied for any integer $\alpha \geq 1$. If (θ_1, p^β) is in R_1, then (8) can be satisfied and $\alpha_* = 1$. If (θ_1, p^β) is in R_2, then (8) can be satisfied and

$$\alpha_* = \left\lceil \frac{\log\ ((1-\theta_1)/\theta_1)}{\log\ ((1-p^\beta)/p^\beta)} \right\rceil \geq 2 \tag{9}$$

where $\lceil x \rceil$ is defined as the smallest integer greater than or equal to x.

Proof: If (θ_1, p^β) is in R_0, Lemma 2 immediately implies that there is no integer $\alpha \geq 1$ for which (8) can be satisfied. If (θ_1, p^β) is in R_1, then it is clear that (8) can be satisfied with $\alpha = 1$. If (θ_1, p^β) is in R_2, then (8) may be rewritten as

$$\alpha \geq \frac{\log\ ((1-\theta_1)/\theta_1)}{\log((1-p^\beta)/p^\beta)}$$

and the result follows immediately since $p^\beta < \theta_1$. ◻

Some examples will now be given.

<u>Example 1</u>: Suppose that $p = 0.9$, $\beta = 10$ and $\theta_1 = 0.8$. In this case, $p^\beta = 0.348$, the pair (θ_1, p^β) is in R_0, and Lemma 4 implies that it is not possible to satisfy the desired reliability constraint.

<u>Example 2</u>: Suppose that $p = 0.95$, $\beta = 10$ and $\theta_1 = 0.95$. In this case, $p^\beta = 0.598$, the pair (θ_1, p^β) is in R_2, and $\alpha_* = 8$.

Example 1 clearly demonstrates the interdependence of the speed and reliability constraints. Our only recourse here would be to choose a different type of computing element: one with a larger p and/or a higher

intrinsic speed allowing for a smaller β. Example 2 shows that satisfaction of the reliability constraint may require a great deal of replication: in this case, 80 computing elements instead of the original 10.

The design approach discussed above may lead to an unacceptably large upper bound on the false alarm probability: in the case of Example 2, $P_{FA,M}(\alpha_*,\beta) = 0.972$. With this consideration in mind, suppose now that we want to design a computing network so that not only will an accepted output be correct with probability at least θ_1 $(0 < \theta_1 < 1)$, but also a correct output will be rejected with probability at most θ_2 $(0 < \theta_2 < 1)$. To satisfy these two requirements, we need to ensure the existence of at least one integer $\alpha \geq 1$ so that both (8) and the following inequality are satisfied (see Lemma 3):

$$1 - p^{\beta(\alpha-1)} \leq \theta_2 . \tag{10}$$

Given p, β, θ_1 and θ_2, it may or may not be possible to satisfy both (8) and (10) with some integer $\alpha \geq 1$. In order to analyze the feasibility issue, we partition the set of all triples $(\theta_1,\theta_2,p^\beta)$ into disjoint subsets S_0, S_1 and S_2 as follows:

$$S_0 = \{(\theta_1,\theta_2,p^\beta) \mid p^\beta < \theta_1 \leq \max\,[0.5,1-\theta_2]\ \} ,$$

$$S_1 = \{(\theta_1,\theta_2,p^\beta) \mid p^\beta \geq \theta_1\ \} ,$$

$$S_2 = \{(\theta_1,\theta_2,p^\beta) \mid \theta_1 > \max\,[0.5,1-\theta_2,p^\beta]\ \} .$$

<u>Lemma 5</u>: If $(\theta_1, \theta_2, p^\beta)$ is in S_0, then (8) and (10) cannot be satisfied simultaneously for any integer $\alpha \geq 1$. If $(\theta_1, \theta_2, p^\beta)$ is in S_1, then (8) and (10) can be satisfied with $\alpha = 1$.

Proof: If $p^\beta < \theta_1$, (8) cannot be satisfied with $\alpha = 1$. If in addition, $\theta_1 \leq 0.5$, then Lemma 2 implies that (8) cannot be satisfied for any integer $\alpha \geq 2$. If $p^\beta < \theta_1 \leq 1-\theta_2$, then (10) cannot be satisfied for any $\alpha \geq 2$ since

$$1 - p^{\beta(\alpha-1)} > 1 - \theta_1 \geq \theta_2 .$$

If $(\theta_1, \theta_2, p^\beta)$ is in S_1, then clearly (8) can be satisfied with $\alpha = 1$, and (10) is always satisfied with $\alpha = 1$. $\square$

Now define

$$g_1(\alpha) = \frac{1}{1 + ((1-\theta_1)/\theta_1)^{1/\alpha}} \quad , \alpha = 2, 3, \ldots ,$$

$$g_2(\alpha) = (1-\theta_2)^{1/(\alpha-1)} \quad , \alpha = 2, 3, \ldots ,$$

$$\alpha_0 = \left\lfloor 1 + \frac{\log (1-\theta_2)}{\log \theta_1} \right\rfloor ,$$

and

$$q_* = \min \{ \max [g_1(\alpha), g_2(\alpha)] \mid \alpha = 2, 3, \ldots, \alpha_0 \} ,$$

where $\lfloor x \rfloor$ is defined as the largest integer less than or equal to x.

<u>Lemma 6</u>: Suppose that the triple $(\theta_1, \theta_2, p^\beta)$ is in S_2. If $p^\beta < q_*$, then

(8) and (10) cannot be satisfied for any integer $\alpha \geq 1$. If $p^{\beta} \geq q_*$, then there is at least one integer $\alpha \geq 2$ for which (8) and (10) are satisfied. Furthermore, $q_* > 0.5$.

Proof: First note that since $(\theta_1, \theta_2, p^{\beta})$ is in S_2,

$$2 < 1 + \frac{\log(1-\theta_2)}{\log \theta_1} < \infty$$

and thus $2 \leq \alpha_0 < \infty$ and q_* is well defined. Furthermore, $\theta_1 > 0.5$ implies that $\theta_1 > g_1(2)$, and $\theta_1 > 1-\theta_2$ implies that $\theta_1 > g_2(2)$. Thus $\theta_1 > q_*$.

Since $p^{\beta} < \theta_1$, it is clear that (8) cannot be satisfied with $\alpha = 1$. For every $\alpha > \alpha_0$, we have

$$\alpha > 1 + \frac{\log(1-\theta_2)}{\log \theta_1}$$

which can be rewritten as

$$1-\theta_1^{(\alpha-1)} > \theta_2 .$$

Therefore

$$1 - p^{\beta(\alpha-1)} > 1-\theta_1^{(\alpha-1)} > \theta_2$$

and (10) cannot be satisfied. Now let $p^{\beta} < q_*$. For each α in the interval $[2,\alpha_0]$, either $p^{\beta} < g_1(\alpha)$, in which case (8) cannot be satisfied, or else $p^{\beta} < g_2(\alpha)$, in which case (10) cannot be satisfied. Thus, if $p^{\beta} < q_*$, (8) and (10) cannot be satisfied for any integer $\alpha \geq 1$.

It is clear from the definition of q_* that there is at least one integer

$\bar{\alpha}$ in the interval $[2,\alpha_0]$ for which

$$q_* = \max\,[g_1(\bar{\alpha}), g_2(\bar{\alpha})]\ .$$

Thus, if $p^\beta \geq q_*$, then $p^\beta \geq g_1(\bar{\alpha})$ and $p^\beta \geq g_2(\bar{\alpha})$, which implies that (8) and (10) are satisfied with $\alpha = \bar{\alpha}$.

If $q_* \leq 0.5$, then (8) cannot be satisfied for $p^\beta = q_*$. This contradicts what has just been proved, and therefore we can conclude that $q_* > 0.5$.

□

Lemmas 5 and 6 show that the conditions under which replication and error detection are useful for meeting constraints (8) and (10) are quite restrictive: we need to have $\theta_1 > 0.5$, $\theta_1 > 1-\theta_2$ and p^β in the interval $[q_*,\theta_1)$.

If (8) and (10) can be satisfied, the most efficient design is obtained when α is chosen to be the smallest integer α_{**} that satisfies (8) and (10). The next lemma characterizes α_{**}.

Lemma 7: If $(\theta_1,\theta_2,p^\beta)$ is in S_1, then $\alpha_{**} = 1$. If $(\theta_1,\theta_2,p^\beta)$ is in S_2 and $p^\beta \geq q_*$, then

$$\alpha_{**} = \left\lceil \frac{\log\ ((1-\theta_1)/\theta_1)}{\log\ ((1-p^\beta)/p^\beta)} \right\rceil .$$

Proof: The first part of the lemma follows immediately from Lemma 5. Now let $(\theta_1,\theta_2,p^\beta)$ be in S_2 with $p^\beta \geq q_*$. Rewrite (8) and (10), respectively, as

$$\alpha \geq \frac{\log\ ((1-\theta_1)/\theta_1)}{\log\ ((1-p^\beta)/p^\beta)}\ ,$$

$$\alpha \lesssim 1 + \frac{\log\ (1-\theta_2)}{\log\ p^{\beta}}\ .$$

Lemma 6 implies that there is at least one integer α satisfying both inequalities, and the result follows. ◻

We will now present some examples that illustrate the preceding lemmas.

Example 3: Suppose that $p = 0.95$, $\beta = 10$, $\theta_1 = 0.8$ and $\theta_2 = 0.1$. In this case, $p^{\beta} = 0.598$, the triple $(\theta_1,\theta_2,p^{\beta})$ is in S_0, and Lemma 5 implies that it is impossible to satisfy the reliability constraints.

Example 4: Suppose that $p = 0.98$, $\beta = 10$, $\theta_1 = 0.9$ and $\theta_2 = 0.2$. In this case, $p^{\beta} = 0.817$, the triple $(\theta_1,\theta_2,p^{\beta})$ is in S_2, $q_* = 0.8$ and Lemma 7 implies that $\alpha_{**} = 2$.

4. CONCLUSION

The approach used in this paper has been based on three principles: (i) one should distinguish between hardware faults in a computing network and incorrect results produced by the network; (ii) one should assume as little as possible about the fault mechanism since in general one knows very little about it; (iii) one should use only those quantities that have some chance of being experimentally measured. These considerations rule out, in particular, the use of a failure probability distribution [BAR65]. They also lead us to a worst case analysis.

Clearly, if one does not adhere to these principles and is willing to make stronger, more optimistic, assumptions, then error detection based on strict redundancy will look more powerful. For example, in view of Equation (6), it is clear that if, in addition to Hypotheses 1 and 2, we assume that when all cluster outputs are incorrect they are not all identical, then $P(B_\alpha(\alpha,\beta)$ and $b=0) = 0$, and thus $P_{CD}(\alpha,\beta) = 1$. In this case $P_{FA,M}(\alpha,\beta)$ is minimized when we use the smallest possible number of clusters, namely $\alpha = 2$.

Alternatively, if in addition to Hypotheses 1 and 2, we assume that the number of faulty computing elements is at most ξ, then the choice of any $\alpha \geq \xi+1$ ensures that at least one computing cluster always produces the correct output. In this case, $P(B_\alpha(\alpha,\beta)) = 0$, which implies $P(B_\alpha(\alpha,\beta)$ and $b=0) = 0$, and we again obtain $P_{CD}(\alpha,\beta) = 1$. The upper bound on the false alarm probability is then minimized by choosing $\alpha = \xi+1$.

REFERENCES

AVI78 Avizienis, A., Fault-Tolerance: The Survival Attribute of Digital Systems, Proc. IEEE, Vol. 66, No. 10, October 1978, pp. 1109-1125.

BAR65 Barlow, R. E. and Proschan, F., Mathematical Theory of Reliability, John Wiley, New York, New York, 1965.

NEU63 Von Neumann, J., Collected Works, Taub, A. H., ed., Vol. 5, MacMillan, New York, New York, 1963.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER JHU/EECS - 83/14	2. GOVT ACCESSION NO. AD-A134 319	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Analysis of an Error Detection Scheme for Parallel Computations		5. TYPE OF REPORT & PERIOD COVERED Technical
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Gerard G. L. Meyer Howard L. Weinert		8. CONTRACT OR GRANT NUMBER(s) N00014-81-K-0813
9. PERFORMING ORGANIZATION NAME AND ADDRESS The Johns Hopkins University Baltimore, MD 21218		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, VA 22217		12. REPORT DATE November 15, 1983
		13. NUMBER OF PAGES 19
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Error Detection - Probability of False Alarm - Parallel Computations

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

We determine the conditions under which an error detection scheme based on strict redundancy can be used to increase confidence in the results of parallel computations. This study shows that the issues of speed and reliability of parallel processors are interdependent and must be considered jointly at the design stage.

END

FILMED

12-83

DTIC